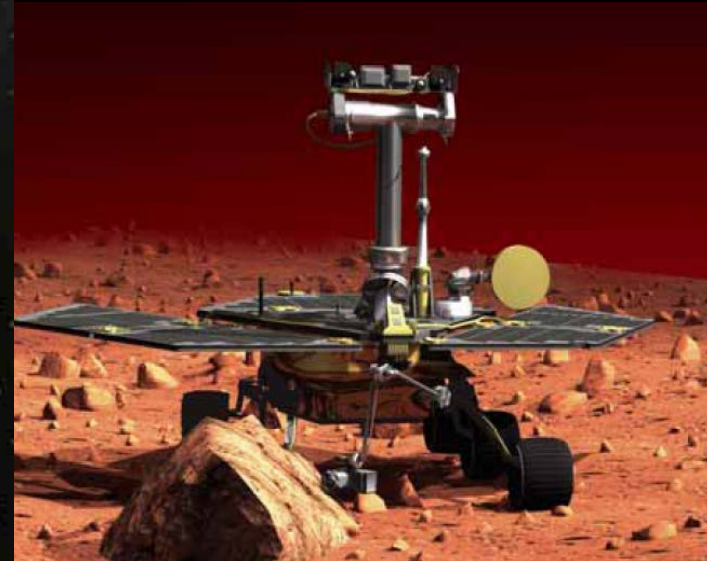
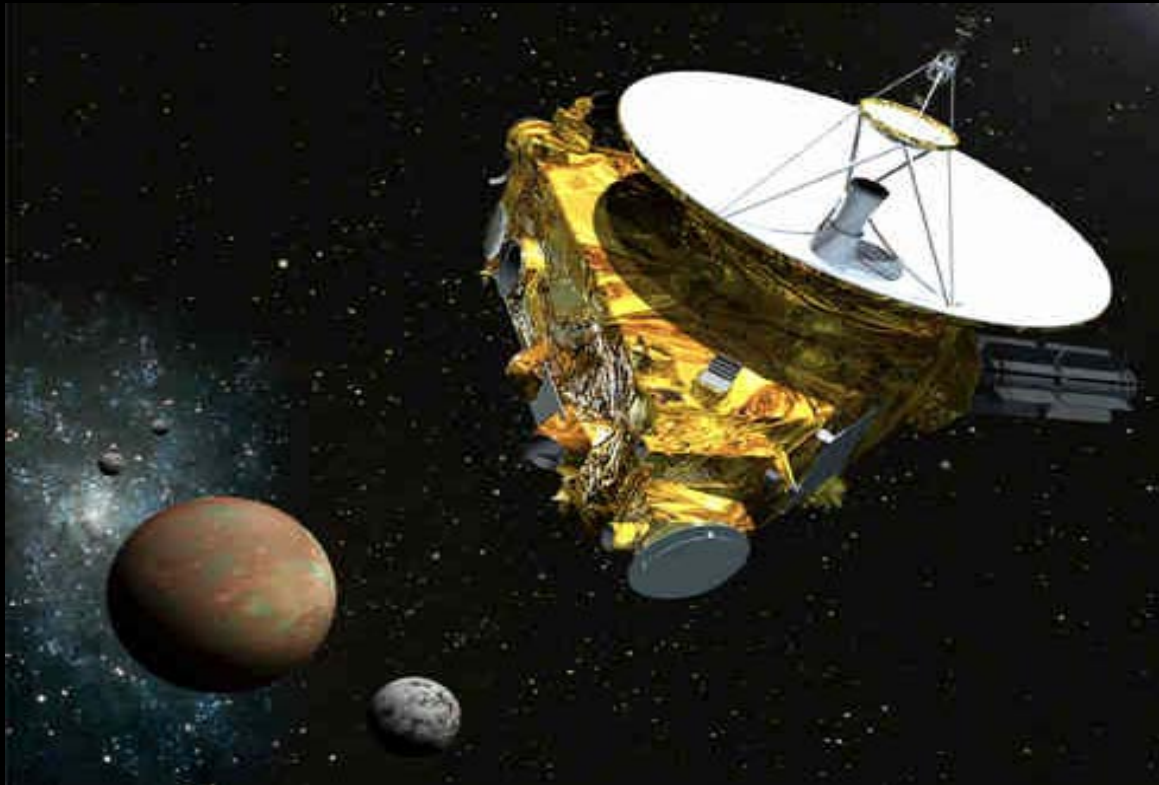




Susceptibility of Redundant Versus Singular Clock Domains Implemented in SRAM-Based FPGA TMR Designs



Melanie Berg, AS&D in support of NASA/GSFC

Melanie.D.Berg@NASA.gov

Keh LaBel, NASA/GSFC

Jonathan Pellish, NASA/GSFC

Acronyms

- Combinatorial logic (CL)
- Design under analysis (DUA)
- Device under test (DUT)
- Distributed triple modular redundancy (DTMR)
- Edge-triggered flip-flops (DFFs)
- Field programmable gate array (FPGA)
- Global triple modular redundancy (GTMR)
- Hardware description language (HDL)
- Input – output (I/O)
- Linear energy transfer (LET)
- Mean time to failure (MTTF)
- Operational frequency (f_s)
- Radiation Effects and Analysis Group (REAG)
- Single Error Correct Double Error Detect Single event functional interrupt (SEFI)
- Single event effects (SEEs)
- Single event transient (SET)
- Single event upset (SEU)
- Single event upset cross-section (σ_{SEU})
- Static random access memory (SRAM)
- Static timing analysis (STA)
- Triple modular redundancy (TMR)

Problem Statement



- Triple modular redundancy (TMR) can be implemented in a variety of topologies.
- This presentation focuses on the trade-offs between implementing TMR with:
 - Multiple clock domains (one clock per TMR domain):
i.e., global TMR (GTMR) and
 - A single clock shared across TMR domains:
i.e., distributed TMR (DTMR).
- For many organizations, GTMR is the mitigation strategy of choice because of its redundant clock topology.
- However, as FPGA devices become larger and more complex, clock skew between separate domains is increasing and becoming impossible to control.
- Unfortunately, mismanaged clock skew can cause timing violations or circuit race conditions in synchronous designs.

Race conditions from clock skew weaken mitigation and can cause system malfunction!



Abstract

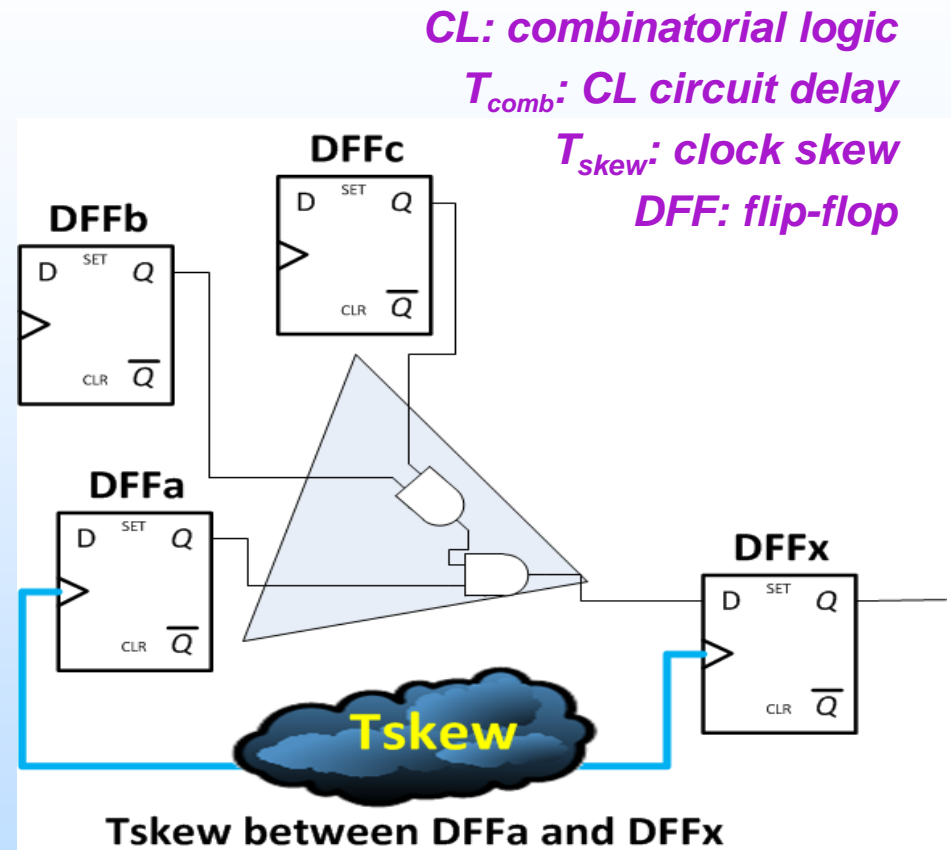
We present the challenges that arise when using redundant clock domains due to their clock-skew. Radiation data show that a singular clock domain (DTMR) provides an improved TMR methodology for SRAM-based FPGAs over redundant clocks.



Clock Skew

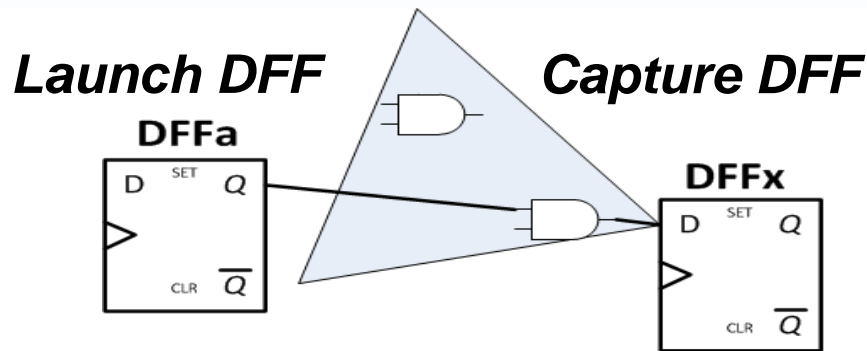
Clock Skew within One Clock Domain

- A clock domain is defined as a group of circuitry that is connected to the same clock tree.
- The clock tree only feeds DFF clock pin inputs.
- It is mandatory to balance the clock tree domain so that all connected DFFs receive the controlling clock edge at virtually the same moment in time.



The difference in time of a clock edge's arrival at one DFF with respect to its arrival at another DFF is defined as clock skew (T_{skew}).

Synchronous Data Capture



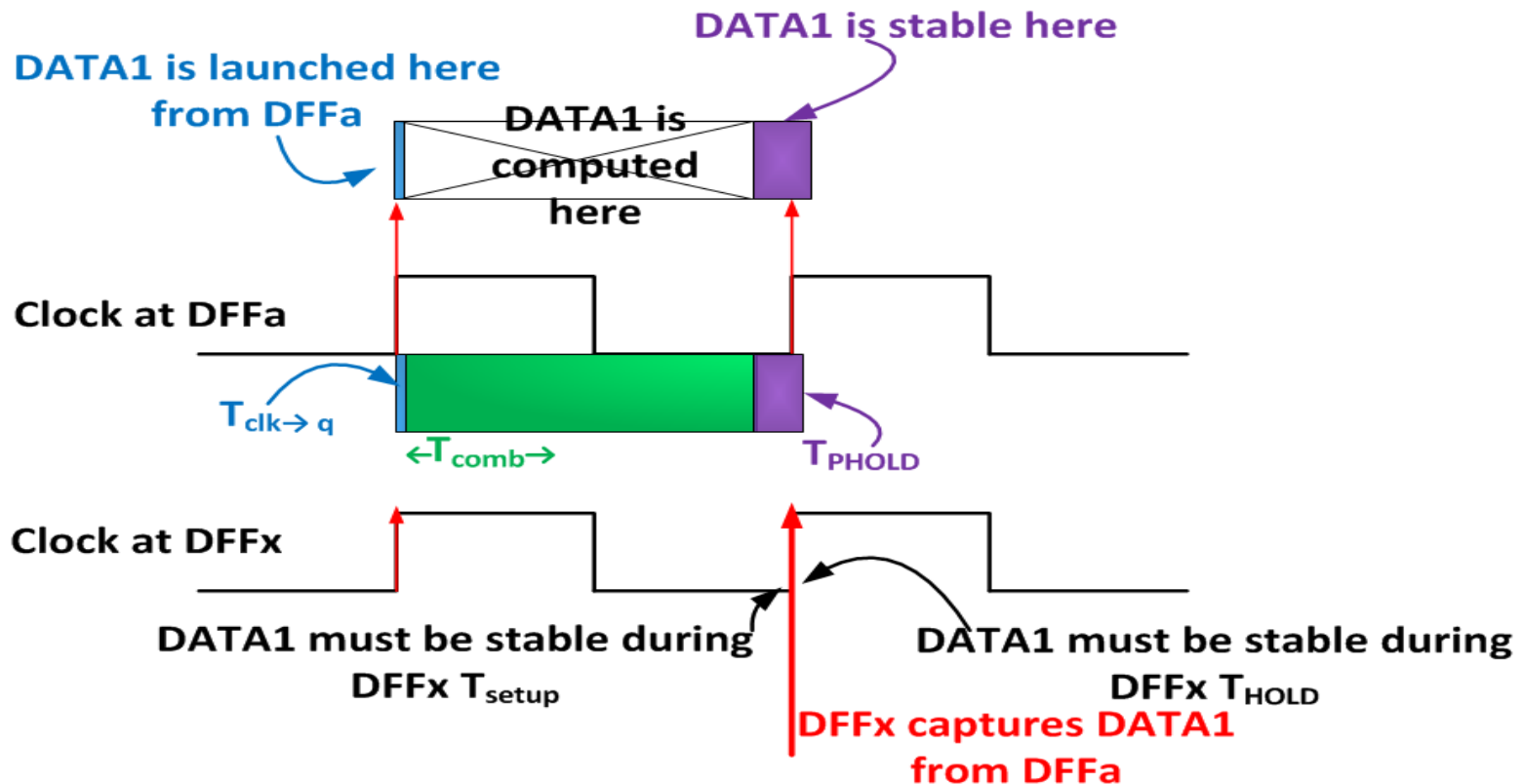
T_{comb} : combinational logic delay.

$T_{clk \rightarrow q}$: delay of data output from DFF.

T_{PHOLD} : Path hold time.

T_{setup} : Data stable time prior to clock edge.

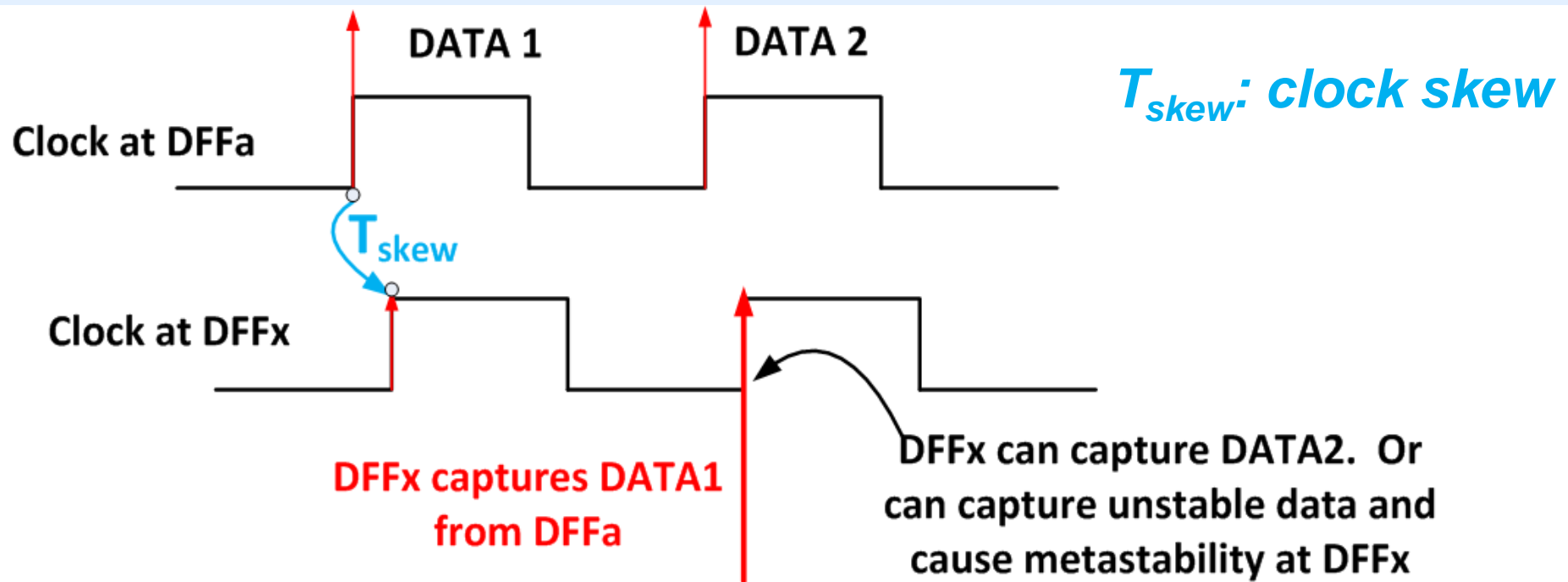
T_{hold} : Data stable time post clock edge.



Positive Skew and Data Capture

In a system with positive skew, there is a possibility that:

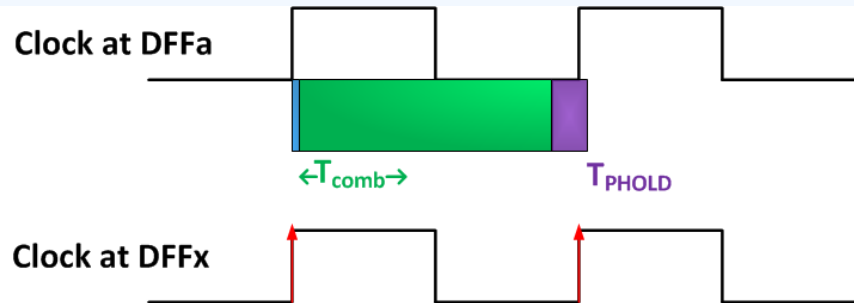
- DFF_x will capture the wrong data (cycle ahead); or
- DFF_x will capture while data is changing (metastability).



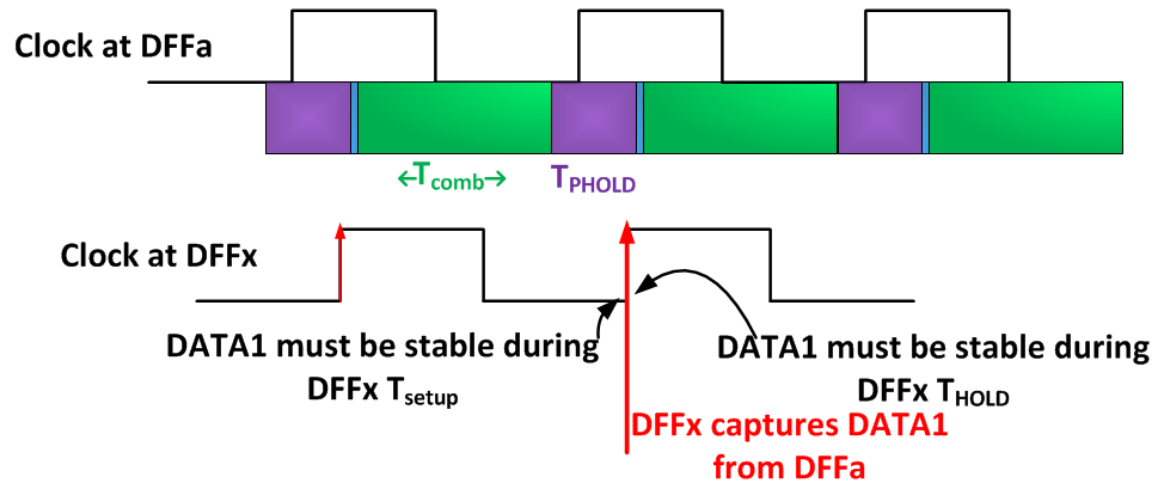
Positive Skew and T_{PHOLD}

- T_{PHOLD} is elongated to accommodate positive clock skew.
- T_{PHOLD} elongation takes time away from T_{comb} ... T_{comb} is shortened.
- Might violate timing constraints.

No
Skew



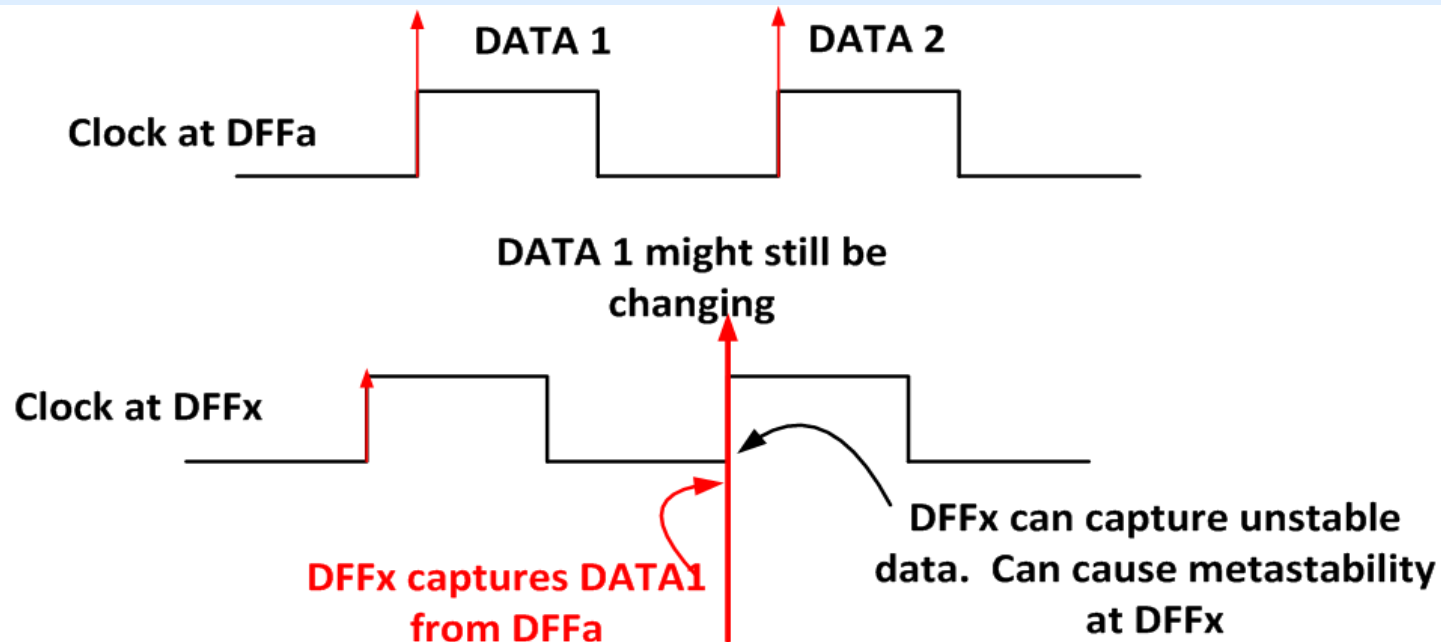
Positive
Skew



Negative Skew and Data Capture

In a system with negative skew, there is the possibility that data can be captured during its computation time.

- T_{setup} is violated.
- This can cause metastability.
- Data is invalid.

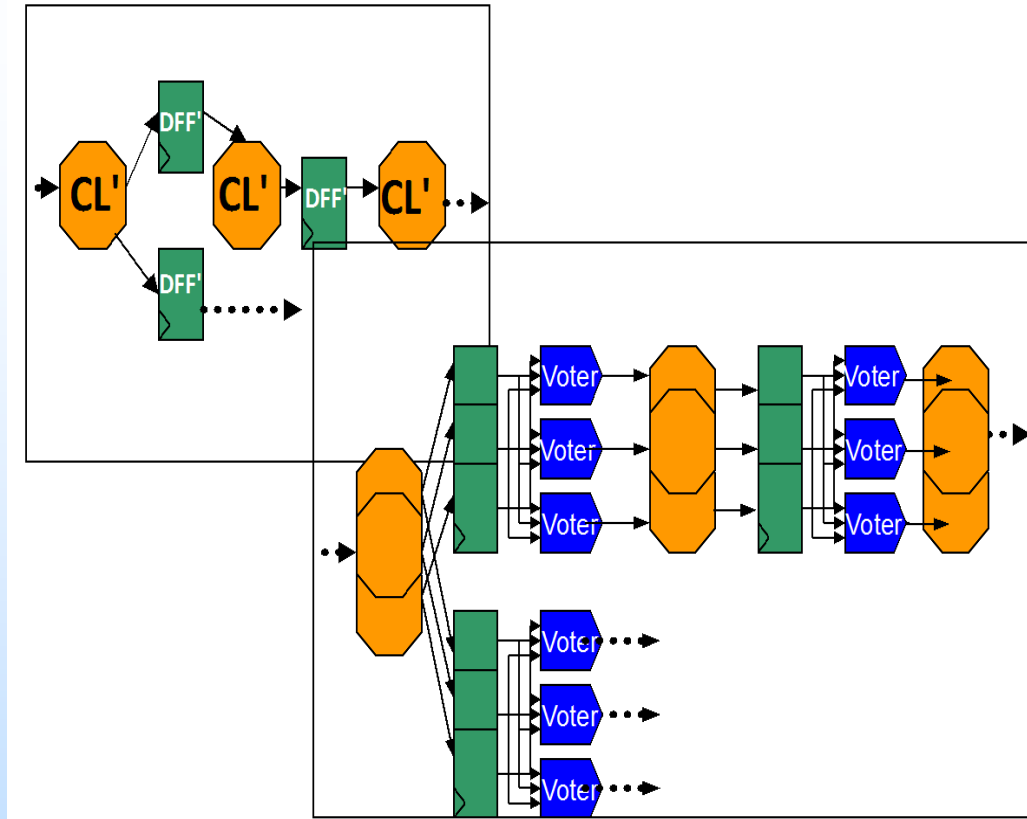




Triple Modular Redundancy (TMR)

DTMR and GTMR Topologies

- With DTMR and GTMR all circuits are triplicated.
- Voters are placed after the internal flip-flops (DFFs).
- DTMR: only **one clock** per TMR domain.
- GTMR: **Three separate clocks** per TMR domain.



GTMR violates synchronous design protocol because of clock skew and sharing data across clock domains without synchronization.

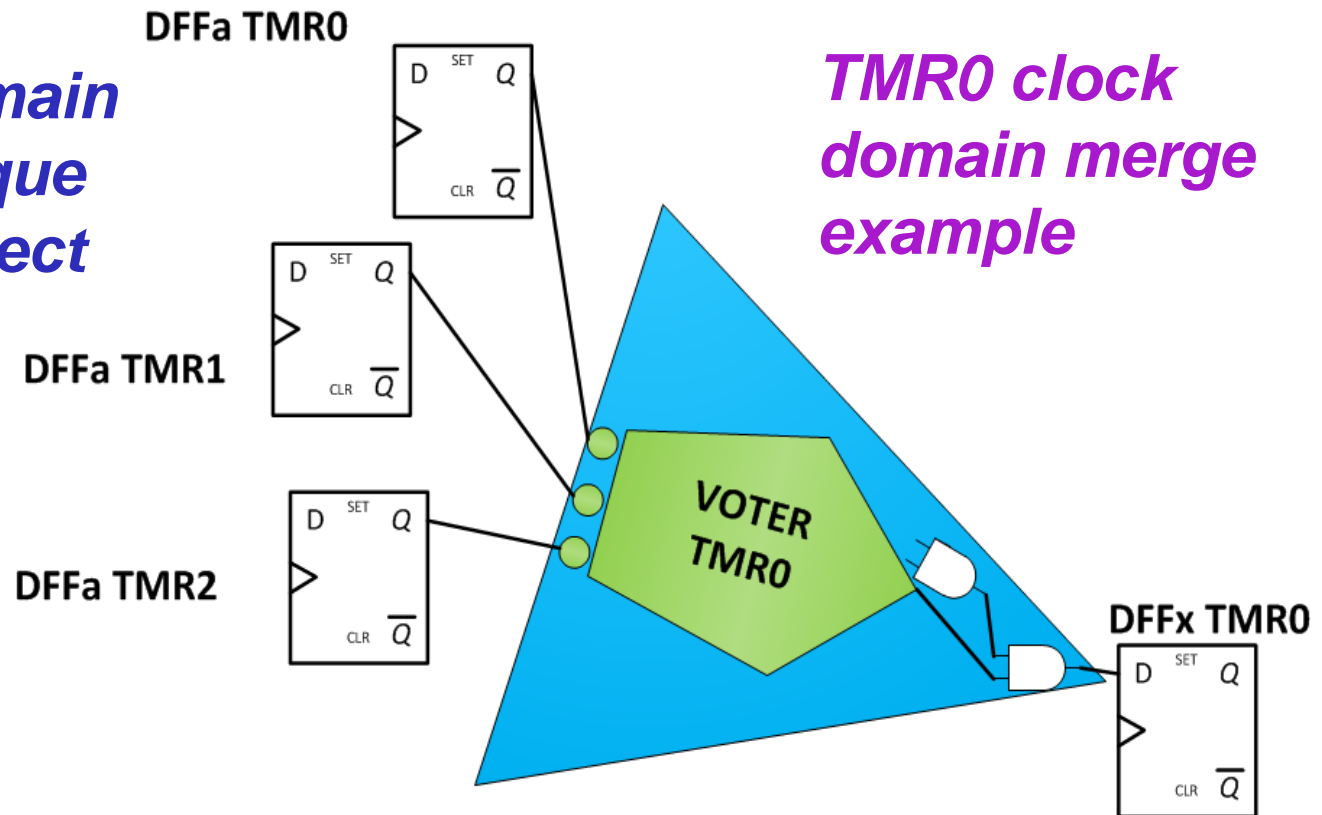


Challenges of GTMR System Implementation

GTMR: Multiple Clock Domains Exacerbate Skew

- Three separate clock domains merge into one clock domain at each voter.
- For GTMR, this is replicated for each clock domain.

Each clock domain will have a unique skew with respect to DFFx.



Complete violation to synchronous design rules.

System Implementation: Sources of Clock Skew



- **Board Level:**
 - One board clock source (oscillator): routes from board clock source must be the same length to FPGA clock inputs.
 - Three board clock sources: Don't!
- **Internal to the FPGA:**
 - Clock pin to clock tree routing differences,
 - Skew within a single clock tree, and
 - GTMR has additional skew from use of different clock trees.

GTMR Skew Management in Various FPGA Devices



- **Board level and routing skew can be managed. However, clock skew within a single tree and between different trees is based on the manufacturer's product and can be difficult or impossible to control.**
- **Clock skew was less of a problem in smaller Xilinx FPGA devices (e.g., Virtex 5 and smaller).**
- **Clock skew is now a challenge in the larger family of Xilinx devices (e.g., 7 series and above).**
 - **More skew within one clock tree (especially as distance between DFFs increase).**
 - **More skew between separate clock trees.**
- **GTMR was never able to be implemented in the Microsemi/Actel FPGA product lines because there is too much skew between clock trees.**

Detection of T_{skew} with GTMR

- **GTMR T_{skew} is difficult to detect due to the following:**
 - Many static timing analysis (STA) tools do not accurately report hold time violations across clock domains – hence the user can be unaware T_{skew} exists.
 - T_{skew} can be temperature and voltage related. Hence, a design can work during ground testing yet have failures during operation in its target environment.
 - In the presence of clock skew, usually two out of three of the domains are in sync. Hence the design will appear to operate normally.
 - Not all nodes will contain the same skew:
 - Some nodes may have positive skew,
 - Some nodes may have negative skew, and
 - Some nodes may have negligible skew.
- **Due to state space explosion, fault injection and simulation will not provide sufficient coverage.**

T_{skew} System Effects

- **Significantly large T_{skew} :** can cause one domain to always be out of sync with the other two domains.
- **Marginal T_{skew} :** can cause metastable circuits.
- **Variable T_{skew} :** can cause pockets of T_{skew} such that some portions of the circuit contain:
 - Positive T_{skew}
 - Negative T_{skew} , and
 - Negligible T_{skew} .
- **As the designer decreases overall T_{skew} , (e.g., via board level and routing management) pockets of T_{skew} start to exist.**
- **This is more prominent in large FPGA devices such as the Xilinx 7-series.**



Accelerated Heavy Ion Testing



Accelerated Radiation Testing: DTMR versus GTMR

- **Device under test (DUT): Xilinx Kintex-7 FPGA (XC7K325T).**
- **The base design (DUA) was the counter-array created by NASA Electronics Parts and Packaging (NEPP) Program.**
- **The counter-array DUA had three versions based on its inserted TMR scheme:**
 - **No TMR,**
 - **GTMR, and**
 - **DTMR.**
- **The TMR DUAs were physically partitioned across TMR domains in order to reduce shared resources (single points of failure).**

Considerations Taken Prior to and during Testing



- **TMR topologies for the DTMR and GTMR designs were analyzed to verify that the redundancy was implemented correctly.**
- **Both DTMR and GTMR were partitioned equally the same.**
- **Major difference is that DTMR has one shared clock and GTMR has three separate clocks.**
- **Multiple bit upsets (MBUs) should not make a significant difference in this testing because each should be statistically equally likely to fail due to MBUs.**

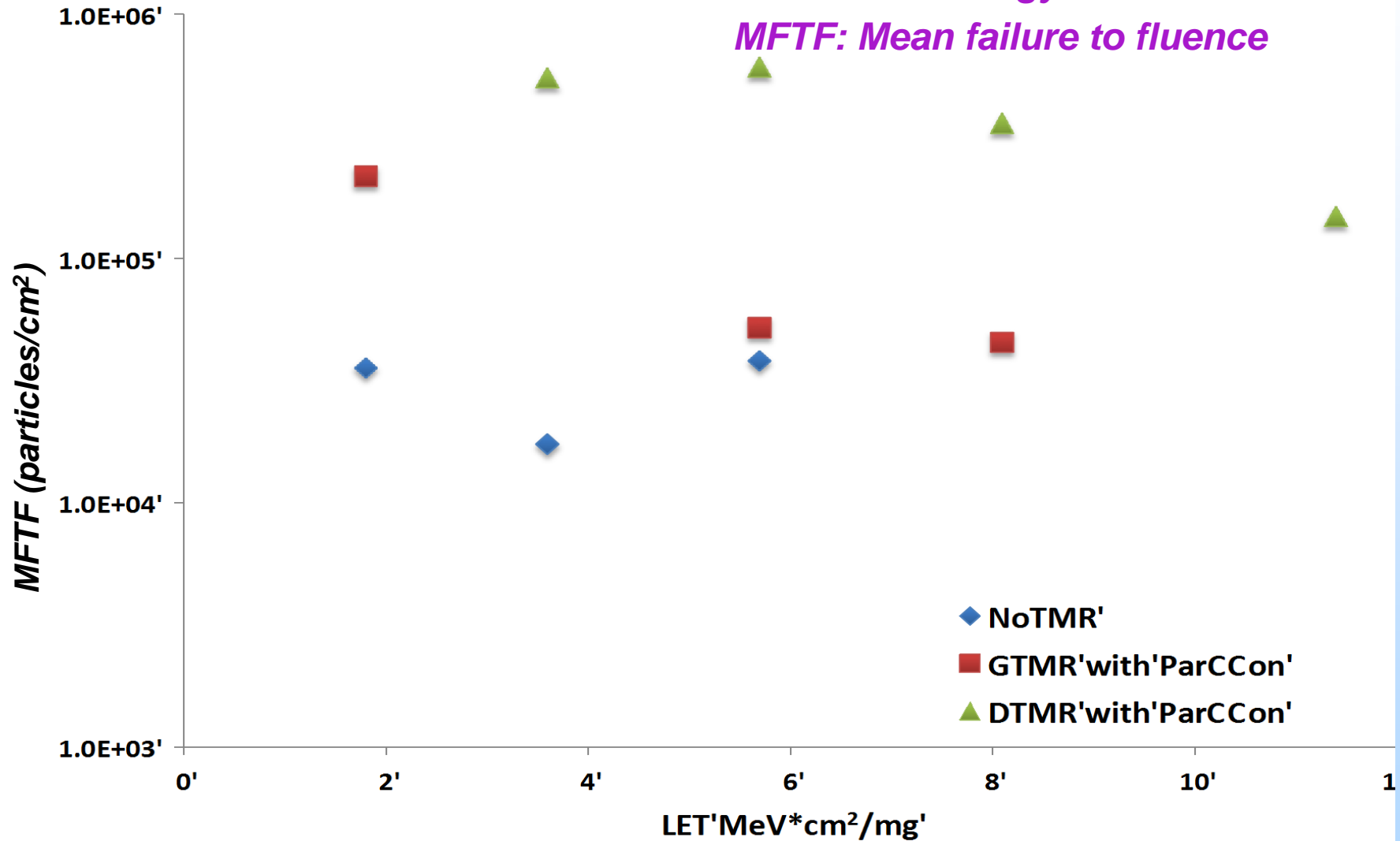
DUA Failures

- Failure is referenced at the system level not at the component level.
- Hence, SEUs can occur and disrupt components; however, if the next state is the expected next state, then no system failure exists.
- In a TMR'd design, system failures can occur from:
 - Single configuration bit-SEUs that control shared resources that cross TMR domains;
 - Multiple configuration bit-SEUs that can span across TMR domains;
 - GTMR architectures with race conditions.
 - Single event functional interrupts (SEFIs)

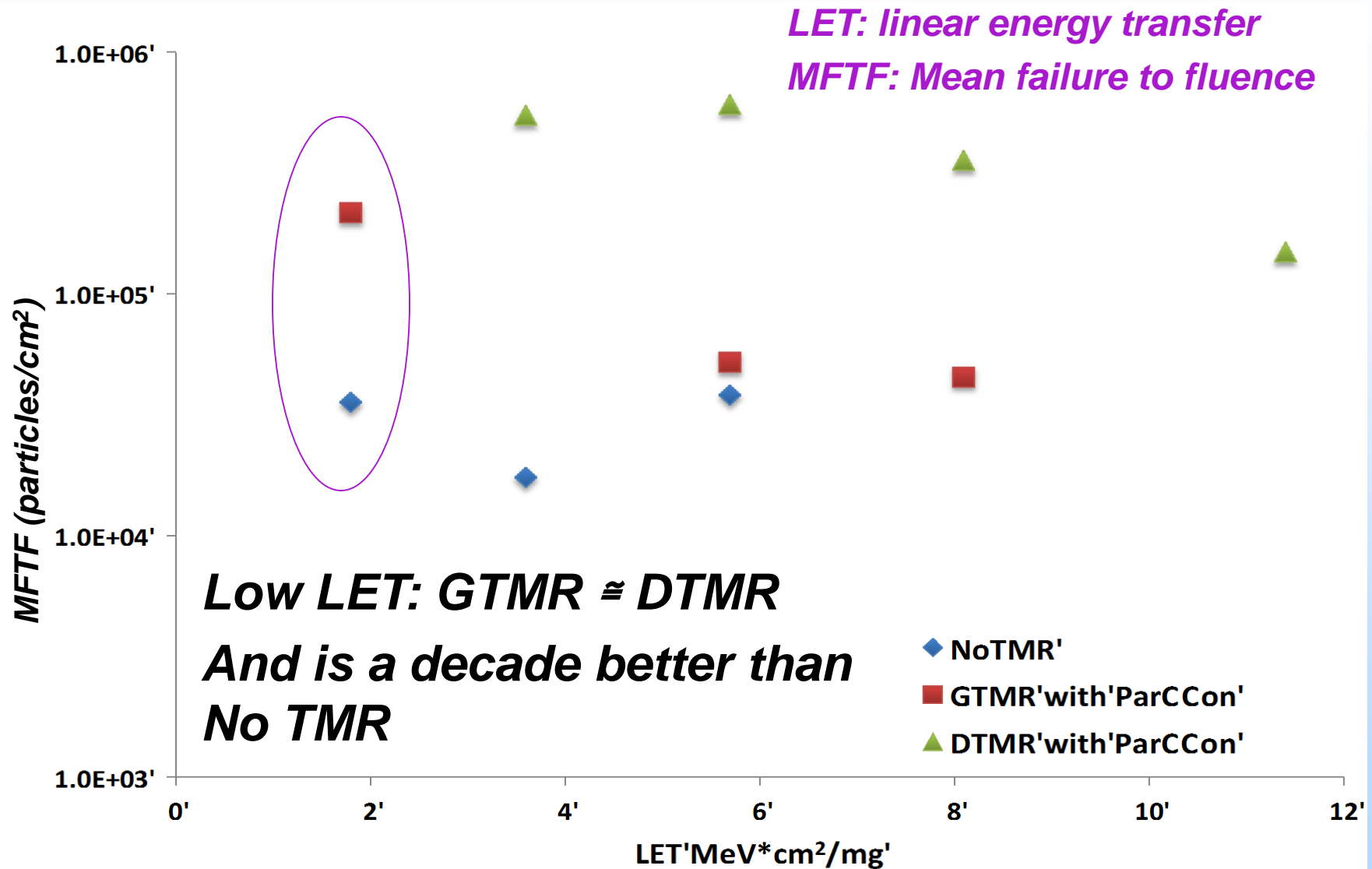
Heavy-Ion Results

LET: linear energy transfer

MFTF: Mean failure to fluence

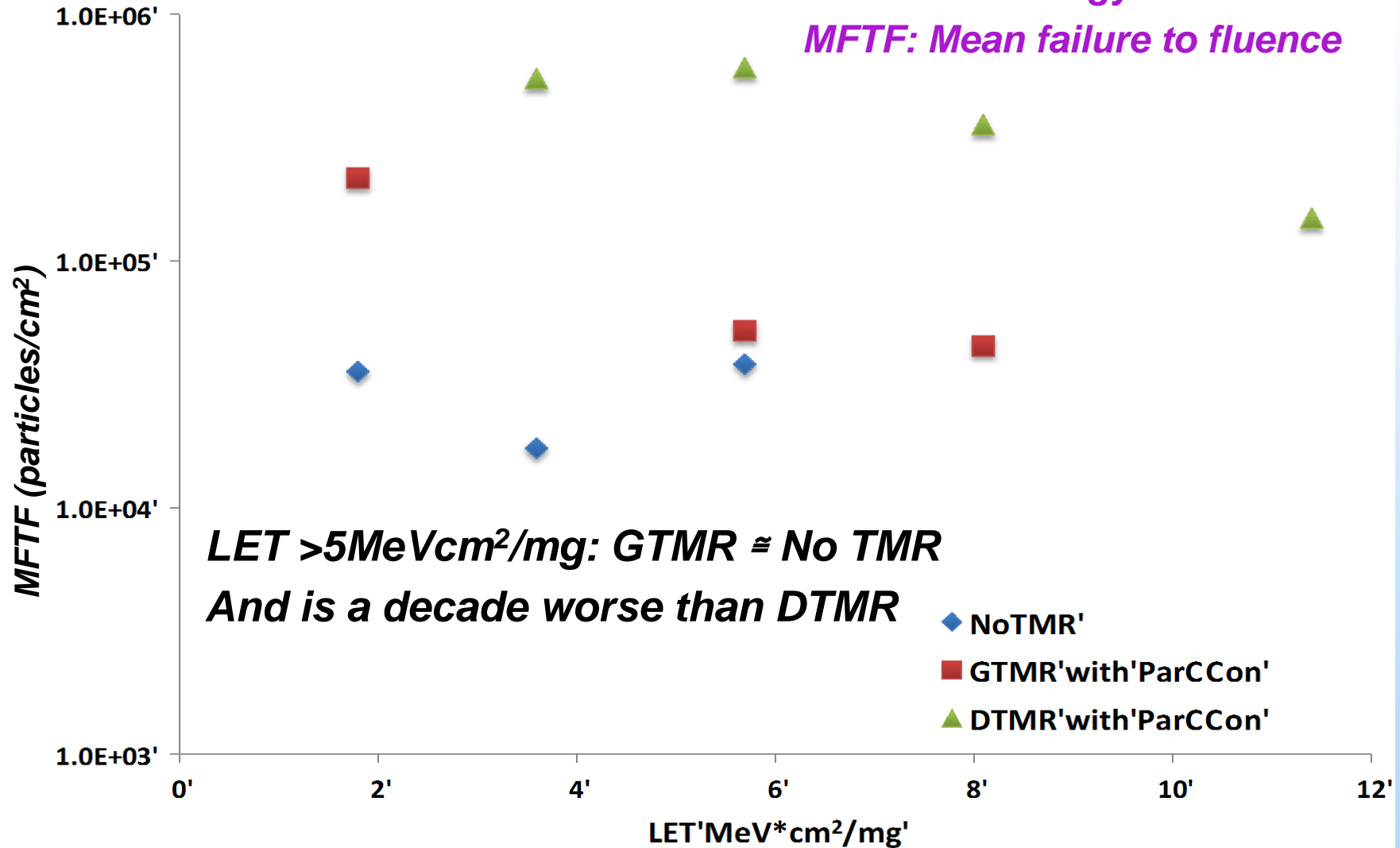


Heavy-Ion Results: Low LETs



Heavy-Ion Results: Higher LETs

LET: linear energy transfer
MFTF: Mean failure to fluence





It Is A Clock Skew Problem

- **If results were due to significant clock skew, one GTMR clock domain would always be out of sync and GTMR would always have results similar to No TMR.**
- **Because GTMR has results near DTMR at low LET and approach No TMR as LET increases, suggest that the failures are mostly due to clock skew.**
- **MBUs are not considered a significant source of failure because both systems are partitioned in the same manner.**

Conclusion (1)

- Theoretically, GTMR should be the strongest TMR mitigation scheme.
- For this reason, it has been suggested as the TMR strategy of choice for SRAM-based FPGAs.
- However, the uncontrollable clock skew between GTMR clock domains can cause race conditions that inevitably weaken GTMR mitigation.
- For small (less complex) designs implemented in FPGAs that contain clock trees with minimal T_{skew} , GTMR can be realizable.



Conclusion (2)

- **As device and design area increase, as with modern devices such as the Xilinx Kintex-7, GTMR clock skew also increases.**
- **The increase in skew increases the potential for race conditions.**
- **Some race conditions can be uncontrollable and unrecognizable by manufacturer-supplied design tools.**
- **Consequently, Kintex-7 GTMR versus DTMR heavy-ion data show that GTMR is an ineffective and unreliable mitigation solution.**
- **In conclusion, we suggest that DTMR is a more applicable TMR strategy for larger commercial SRAM-based FPGA devices.**



Acknowledgements

- *Some of this work has been sponsored by the NASA Electronic Parts and Packaging (NEPP) Program and the Defense Threat Reduction Agency (DTRA).*
- *Thanks is given to the NASA Goddard Radiation Effects and Analysis Group (REAG) for their technical assistance and support. REAG is led by Kenneth LaBel and Jonathan Pellish.*

Contact Information:

Melanie Berg: NASA Goddard REAG FPGA

Principal Investigator:

Melanie.D.Berg@NASA.GOV